Insights into the Lottery Ticket Hypothesis and the Iterative Magnitude Pruning Algorithm

Tausifa Jan Saleem*, Ramanjit Ahuja**, Surendra Prasad, Brejesh Lall
Bharti School of Telecommunication Technology and Management
Indian Institute of Technology Delhi, Delhi, India

***Abstract*—The Lottery Ticket Hypothesis [1] for deep neural networks emphasizes the importance of initialization used to re-train the sparser networks obtained using the iterative magnitude pruning process. An explanation for why the specific initialization proposed by the lottery ticket hypothesis tends to work better in terms of generalization (and training) performance has been lacking. Moreover, the underlying principles in iterative magnitude pruning, like the pruning of smaller magnitude weights and the role of the iterative process, lack full understanding and explanation. In this work, we attempt to provide insights into these phenomena by empirically studying the volume/geometry and loss landscape characteristics of the solutions obtained at various stages of the iterative magnitude pruning process.**

***Index Terms*—Deep neural networks, Neural network pruning, Lottery ticket hypothesis, Iterative magnitude pruning, Loss landscape visualization.**

I. Introduction

Network Pruning [2] is a widely used sparsification technique in Edge AI to reduce the model size and energy consumed for achieving efficient inference. It has been observed that the pruned models do not perform well without re-training, and re-training the sparser networks from start (with random initialization) is difficult [3], [4]. However, Frankle and Carbin [1] demonstrated that there exists a sub-network, which if trained from the start, reaches the accuracy of the original network. More formally, their hypothesis (called the *Lottery Ticket Hypothesis (LTH)*) states that a randomly initialized dense neural network contains a sub-network which when trained from this initialization can match the test accuracy of the original network after training for at most the same number of iterations. Empirical demonstration of LTH consists of a procedure called *Iterative Magnitude Pruning (IMP)* [1], [5]. IMP has been successful in producing highly-sparse networks that have matching performance, but the underlying principles like the role of specific initialization proposed by LTH, pruning of smaller magnitude weights, and the role of the iterative process are poorly understood. Many research works related to LTH have been published since its advent to demystify the mechanisms and principles governing it: [6]–[18].

In this work, we attempt to provide insights about LTH and IMP by studying the *training loss* landscape characteristics and volume/geometry of the IMP solutions at different iterations. The closest work to ours is [19] which has studied the geometry of the *error (test error)* landscape to answer several questions about LTH and IMP. Although training loss and test error are correlated with each other, studying the training loss landscape instead of the error landscape makes more sense for developing an understanding of model training because the stochastic gradient descent(SGD) training algorithm navigates the loss landscape and not the error landscape.

Contributions. The study presents new insights into LTH and IMP, shedding light on the role of LTH initialization and the mechanisms that allow pruned networks to retain high performance despite significant sparsity. The insights are arrived at through specially designed experiments using a widely used network (Resnet-20) and a benchmark dataset (CIFAR-10). In the interest of bringing out the model behaviour clearly and keeping a complex analysis tractable, this work does not cover experiments with more advanced architectures which would be the subject of future work. The contributions are as follows:

- We demonstrate that the loss landscape contains special solutions, which generalize very well but have very small volume in the unpruned weight space (therefore, not discoverable) but have larger volumes in the pruned space.
- We provide insight into the role played by the specific LTH initialization.
- We provide insight into why pruning needs to be iterative, and why a one-shot approach does not work well (relatively).
- We show that there exists a barrier between different IMP solutions in the *loss* landscape implying that they are not strictly linearly connected, and hence not easily discoverable by SGD.
- We show that IMP solutions (during the iterative process) obtained using rewinding lie in the same (connected) sub-level set in the loss landscape.
- We argue why weight fine-tuning does not work at par with rewinding.

II. Background Information

Fine-tuning and rewinding. IMP consists of a number of pruning and re-training cycles with each re-training cycle preceded by either weight fine-tuning or rewinding. Conventionally, IMP used fine-tuning or rewinding with random initialization, while LTH introduced rewinding with specific

*Corresponding author (tausifa.cstaff@iitd.ac.in). This work was accomplished during the author's affiliation with IIT Delhi. She is currently affiliated with MBZUAI as a Postdoctoral Researcher.

**This work was accomplished during the author's affiliation with IIT Delhi. He is currently affiliated with ON Semiconductor Corp (onsemi) as Member of Technical Staff.

initializations resulting in better performance. Fine-tuning trains the pruned network with a fixed (small) learning rate, using the final values of the remaining weights as the starting point [3], [20]. Rewinding is of two types; weight rewinding and learning rate rewinding. In the case of weight rewinding, the unpruned(remaining) weights are rewound to initialization values used in the last training, and learning rate schedule is also reset. Learning rate rewinding does not rewind weights but resets the learning rate schedule to the one followed by the weight rewinding procedure. Renda et al., [21] demonstrated that fine-tuning does not perform as well as weight rewinding, however, learning rate rewinding performs at par with weight rewinding or even outperforms it in some scenarios.

Iterative pruning versus one-shot pruning. In contrast to iterative pruning and retraining used in IMP, one-shot pruning removes weights in one go to attain the desired sparsity and then retrains. It has been demonstrated in the literature that iterative pruning always outperforms one-shot pruning [1], [21].

Loss landscape of a network. During model training, a loss function of the form:

$$Loss(W) = \frac{1}{n}\Sigma_{i=1}^{n} l(x_i, y_i, W), \tag{1}$$

is minimized, where x_i represents the feature vectors, y_i, the corresponding labels, W, the network parameters, n, the number of samples, and $l(x_i, y_i, W)$ is a function that measures the difference between the predicted label and the actual label. The number of parameters in models is very large; hence, the network loss functions reside in extremely high-dimensional spaces. The plot of training loss ($Loss(W)$) with respect to the network parameters (W) is referred to as the loss landscape. Studying the loss landscape of neural networks is important for understanding their behavior. The loss landscape of under-parameterized models has multiple isolated local minima [22]. The set of solutions of over-parameterized models, on the other hand, is generically a manifold of dimension $m - rn$ [23], where m is the number of parameters, n is the number of input data samples, and r is the number of output classes. This means that the density of minima in the loss landscape of over-parameterized models is very high, and the optimizer converges to one of them irrespective of where it starts at.

Error landscape of a neural network. The variation in the test error of a neural network with respect to the network parameters is referred to as the error landscape of a neural network. It provides insights into the network performance and sensitivity to the changes in the parameters.

Loss sub-level set. Loss sub-level set $S(\epsilon)$ is the set of all points in the weight space for which loss $Loss(W)$ is less than or equal to some desired value ϵ [11] :

$$S(\epsilon) := W \in R^D : Loss(W) \leq \epsilon \tag{2}$$

Role of volume[1] in generalization performance. Flatness is an indicator of performance sensitivity to parameter perturbations. The minimum is flat if small changes to the parameters do not cause misclassifications. On the contrary, the minimum is sharp if small changes to the parameters cause a number of misclassifications, thereby increasing the value of the loss function. A number of studies have focused on establishing the relationship between the flatness/sharpness of minima and their generalization ability [24]–[28]. Huang et al. [29] demonstrated that two types of minima exist in a neural network loss landscape. These are referred to as the so-called good minima and bad minima. Good minima exhibit a small training loss and a small generalization error. Bad minima, too have a small training loss but exhibit a high generalization error. They also studied the qualitative difference in the loss landscape around these minima and observed that the decision boundaries of good minima have wide margins[2] while the decision boundaries of bad minima have very narrow margins. Huang et al. [29] also illustrated that the good minima reside in wide basins[3] that exhibit a large volume in the parameter space, while the bad minima reside in narrow basins that exhibit a much smaller volume. A larger volume also implies a higher probability of hitting the minima by SGD. Volumes of the minima are, therefore, good indicators of their robustness and can provide useful insights [29]. Calculation of the volume of these basins is, however, computationally intractable because the loss function lies in an extremely high-dimensional space. Huang et al. [29] used Monte-Carlo integration (which is computationally intensive) to approximate the volume of these basins. Another approach to estimating the basin volume was given by [30], who demonstrated that the product of top-k positive eigenvalues (λ) of the Hessian at the minimum point could be used to approximate the inverse volume of these basin. More specifically, the inverse volume of the basin ($V^{'}$) can be approximately expressed as:

$$V^{'}(k) := \Sigma_{i=1}^{k} Log(\lambda_i) \tag{3}$$

This is because a large basin implies that the valley around the minimum is flat, which is associated with smaller eigenvalues, and vice-versa.

III. Definitions and Notations.

Sparse sub-networks: Given a dense network with weights W ($W \in \mathbb{R}^D$), a sparse sub-network has weights $m \odot W$, where $m \in \{0,1\}^D$ is a binary mask and $\odot$ is the element-wise product. The sparsity of a mask m, $S(m)$ is the fraction of zeros in the mask.

Notation for IMP solution at level L: We represent the IMP solution (minimum) at level L by $W_{(L)}^{(min\_(L))}$, weights of the dense network at initialization by $W^{(init)}$ and weights of the dense network at rewind-point by $W^{(rewind\_point)}$. Note that all these weights are D-dimensional.

[1]For our discussion here, volume refers to the volume of a solution (minimum) and not the volume of loss sub-level set.

[2]Distance between the class boundary and the data.

[3]Set of points in the neighborhood of a minimum whose loss value is smaller than some cutoff value.

Projection of level L solution on level $(L+1)$: Let $W_{(L+1)}^{Pr(min\_(L))}$ represent the projection of level L solution on level $(L+1)$. It is obtained as $W_{(L+1)}^{Pr(min\_(L))}$=$m_{(L+1)} \odot W_{(L)}^{(min\_(L))}$, where $m_{(L+1)}$ represents the pruning mask at level $(L+1)$. For example, the projection of level 0 solution on level 1 will be represented as $W_{(1)}^{Pr(min\_(0))}$. Note that $S(m_{(L+1)}) > S(m_{(L)})$.

Reverse Projection of level $(L+1)$ solution on level (L): Let $W_{(L)}^{RPr(min\_(L+1))}$ represent the reverse projection of level $(L+1)$ solution on level (L). It is obtained as; $W_{(L)}^{RPr(min\_(L+1))}$=$m_{(L)} \odot W_{(L+1)}^{(min\_(L+1))}$, where $m_{(L)}$ represents the pruning mask at level (L). For example, the reverse projection of level 1 solution on level 0 will be represented as $W_{(0)}^{RPr(min\_(1))}$.

IV. Methodology

We perform experimentation on a ResNet-20 network trained on CIFAR-10 dataset using iterative magnitude pruning with weight rewinding (IMP-WR) for 10 iterations. The unpruned weights are rewound to their values at 2000^{th} training step of the original dense network in each iteration. A plot of training loss and test accuracy at different levels of IMP-WR is presented in Figure 1. It is also useful to compare the

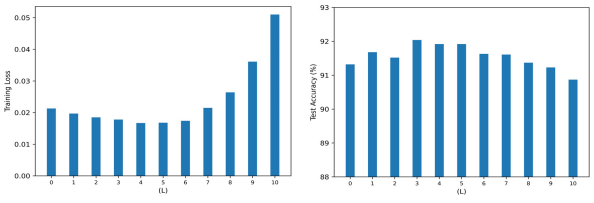

Fig. 1. Training loss and test accuracy at different levels of IMP-WR. **Left:** Training loss. **Right:** Test accuracy.

performance of IMP-WR with other strategies that may either produce networks with a similar structure as that produced with IMP-WR but are initialized differently or produce networks with the same sparsity but different structures. To this end, we apply techniques like one-shot pruning, fine-tuning, random initialization of the pruned network, and random pruning on the aforementioned network. The details of these techniques are summarized below:

One-shot pruned network is obtained by pruning the weights of the trained dense network $W_{(0)}^{(min\_(0))}$ based on magnitude pruning in one go to attain the desired sparsity[4] and then rewinding the unpruned weights to their values at $W^{(rewind\_point)}$ and retraining. We represent the solution obtained using one-shot pruning by $W_{(10)}^{(one\_shot)}$.

Fine-tuned network is obtained by pruning 20% smallest magnitude weights from $W_{(9)}^{(min\_(9))}$ and then re-training the unpruned weights (without rewinding) with a learning rate of 0.001 for 40 epochs. We represent the solution obtained using fine-tuning by $W_{(10)}^{(FT)}$.

Randomly initialized pruned network (represented by $W_{(10)}^{(RIPN)}$) is obtained by pruning 20% smallest magnitude weights from $W_{(9)}^{(min\_(9))}$ and then randomly initializing the unpruned weights and retraining.

Randomly pruned network (represented by $W_{(10)}^{(RPN\_1)}$) is obtained by randomly pruning 20% weights from $W_{(9)}^{(min\_(9))}$ and then rewinding and retraining. We also consider another randomly pruned network ($W_{(10)}^{(RPN\_2)}$) by randomly pruning the weights of $W_{(0)}^{(min\_(0))}$ in one go to attain the desired sparsity and then rewinding and re-training. Note that the only difference between $W_{(10)}^{(RPN\_2)}$ and $W_{(10)}^{(one\_shot)}$ is that while obtaining $W_{(10)}^{(RPN\_2)}$, the weights of $W_{(0)}^{(min\_(0))}$ are pruned randomly whereas while obtaining $W_{(10)}^{(one\_shot)}$, the weights of $W_{(0)}^{(min\_(0))}$ are pruned using magnitude based pruning.

A comparison of training loss and test accuracy between $W_{(10)}^{(min\_(10))}$, $W_{(10)}^{(one\_shot)}$, $W_{(10)}^{(FT)}$, $W_{(10)}^{(RIPN)}$, $W_{(10)}^{(RPN\_1)}$ and $W_{(10)}^{(RPN\_2)}$ presented in Figure 2 shows that $W_{(10)}^{(min\_(10))}$ outperforms the other solutions. To explain the reason behind this, we study the loss landscape of these networks.

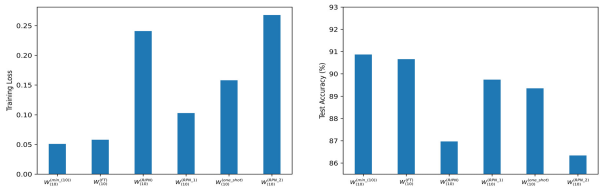

Fig. 2. Comparison of training loss and test accuracy between $W_{(10)}^{(min\_(10))}$, $W_{(10)}^{(one\_shot)}$, $W_{(10)}^{(FT)}$, $W_{(10)}^{(RIPN)}$, $W_{(10)}^{(RPN\_1)}$ and $W_{(10)}^{(RPN\_2)}$. **Left:** Training loss. **Right:** Test accuracy.

V. Results and Findings

In this section, we summarize and explain our major findings based on the conducted experimentation.

A. Result 1: Special solutions with small volume exist.

Huang et al. [29] had proposed that there exist two kinds of minima in a neural network loss landscape: good minima and bad minima. We demonstrate that there also exist another kind of solutions [5] which have good generalization performance but have a (relatively) small volume. This implies that volume is not the only criterion for generalization performance; there is more to it. A careful experimental study leads us to hypothesize that the small volume of these solutions is due to very sharp curvature in certain dimensions, but the coefficients in these dimensions are zero. Their *volume* measure tends to increase when these inferior dimensions are removed (possibly

[4]Same sparsity as that of $W_{(10)}^{(min\_(10))}$.

[5]We can not say for sure that these solutions are true minima, but they lie in the neighbourhood of minima because the gradient of the vast majority of parameters at these points is zero.

via pruning at another point) but is small when considered in the original space. **This makes these solutions almost undiscoverable by SGD in the original space but can be readily discovered in the pruned space.** This is the main result of our study.

Consider two IMP solutions $W_{(L-1)}^{(min\_(L-1))}$ and $W_{(L)}^{(min\_(L))}$ at levels $(L-1)$ and L, respectively. $W_{(L-1)}^{(min\_(L-1))}$ is a baseline for $W_{(L)}^{(min\_(L))}$ because the pruning mask for level L is determined by $W_{(L-1)}^{(min\_(L-1))}$. A comparison of the euclidean distance between $W_{(L)}^{Pr(min\_(L-1))}$ and $W_{(L)}^{Pr(rewind\_point)}$, and the euclidean distance between $W_{(L)}^{(min\_(L))}$ and $W_{(L)}^{Pr(rewind\_point)}$ given in Figure 3 shows that for all L except 2 and 3, $W_{(L)}^{Pr(min\_(L-1))}$ is closer to $W_{(L)}^{Pr(rewind\_point)}$ than $W_{(L)}^{(min\_(L))}$. Despite this, at level (L)

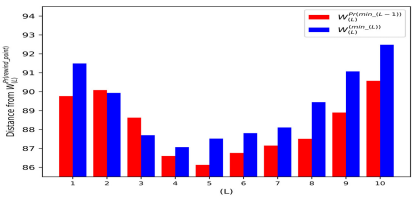

Fig. 3. Distance from $W_{(L)}^{Pr(rewind\_point)}$ to $W_{(L)}^{Pr(min\_(L-1))}$ and to $W_{(L)}^{(min\_(L))}$.

SGD converges to $W_{(L)}^{(min\_(L))}$ and not to $W_{(L)}^{Pr(min\_(L-1))}$. In order to find the underlying reason, we plot the trajectory of SGD for level L and that for level $(L-1)$ projected on level L, which is shown in Figure 4.

It is evident from the figure that the trajectory for level (L) is steeper than for level $(L-1)$ projected on level L. This makes SGD converge to $W_{(L)}^{(min\_(L))}$ and not to $W_{(L)}^{Pr(min\_(L-1))}$. Further, if we compare the volume of basins around $W_{(L)}^{(min\_(L))}$ and $W_{(L)}^{Pr(min\_(L-1))}$ by calculating the product of top-100 positive eigenvalues of the Hessian (of the loss function), the volume[6] of basin around $W_{(L)}^{(min\_(L))}$ is seen to be larger than the volume of the basin around $W_{(L)}^{Pr(min\_(L-1))}$. This is demonstrated in Figure 5 and Figure 6.

Next, if we consider the reverse projection of level (L) solution on level $(L-1)$, $W_{(L-1)}^{RPr(min\_(L))}$ and the level $(L-1)$ solution, $W_{(L-1)}^{(min\_(L-1))}$, the volume of basin around $W_{(L-1)}^{(min\_(L-1))}$ is seen to be larger than the volume of basin around $W_{(L-1)}^{RPr(min\_(L))}$. This is demonstrated in Figure 7 and Figure 8.

And this explains why SGD does not converge to $W_{(L-1)}^{RPr(min\_(L))}$ at level $(L-1)$. To sum up the discussion above, SGD converges to $W_{(L)}^{(min\_(L))}$ at level (L) instead of $W_{(L)}^{Pr(min\_(L-1))}$ because the volume of the basin around $W_{(L)}^{(min\_(L))}$ is much larger than the volume of the basin around $W_{(L)}^{Pr(min\_(L-1))}$, and the path to $W_{(L)}^{(min\_(L))}$ is steeper than the path to $W_{(L)}^{Pr(min\_(L-1))}$. However, in the $(L-1)$ space, the volume of the basin around $W_{(L-1)}^{RPr(min\_(L))}$ is much smaller than the volume of basin around $W_{(L-1)}^{(min\_(L-1))}$. This means that the volume comparison gets flipped in the two spaces, $(L-1)$ space and (L) space.

B. Result 2: Why does the initialization proposed by the lottery ticket hypothesis work well?

The above result also provide a strong clue to the positive role of the initialization proposed by LTH in finding good minima in the sparser weight space. Pruning out the smaller weights leads to a new minimum or a saddle point with a slightly larger training loss value and a smaller volume. Retraining from this point via learning rate rewinding or from a point in close proximity (via weight rewinding) leads SGD to converge to a new minimum, which has a larger volume (compared to the earlier minimum) and a lower training loss. This minimum was not discoverable earlier since it spanned some dimensions where the loss function was steeply increasing and therefore, had an overall smaller volume, and pruning out these dimensions exposed this minimum to the SGD. Hence, the starting minimum *acts as a baseline*, which causes the SGD algorithm to attempt to find a better[7] minimum in the vicinity. Choosing any other initialization point likely places the SGD at a point outside the loss sub-level set and there is no guarantee of finding a better minimum. Figure 9 presents the training loss along a straight line connecting $W_{(9)}^{(min\_(9))}$ (baseline for $W_{(10)}^{(RIPN)}$) and $W_{(10)}^{(RIPN)}$. The figure shows a significant barrier between the two points, demonstrating clearly that these two points lie in different loss sublevel sets. Further, a comparison of top-100 positive eigenvalues of the Hessian at $W_{(10)}^{(RIPN)}$ and $W_{(10)}^{(min\_(10))}$ (Figure 9) shows that $W_{(10)}^{(RIPN)}$ has larger eigenvalues than $W_{(10)}^{(min\_(10))}$, which indicates a smaller volume for the basin around $W_{(10)}^{(RIPN)}$. Hence, random initialization of a pruned network takes SGD out of the sub-level set, and after retraining, converges to a minimum with a smaller volume. The importance of initialization proposed by LTH, therefore, becomes quite apparent.

C. Result 3: Why does one-shot pruning not work well?

Continuing the argument presented in Results 1 and 2, the new (potentially better) minimum that is now exposed to SGD has a larger volume and lower training loss than the earlier pruned minimum which acts as a baseline for this better minimum to be found in the next step. If too many weights are pruned out in a step, it leads to a bigger increase in the training loss and a bigger decrease in volume, thus lowering the baseline for the next solution. If too small a number of weights are pruned out in a step, we risk returning to the same

[6]Smaller the eigenvalues, the smaller their product will be, and the larger would be the volume of the basin around the minimum.

[7]In the worst case SGD may return to the baseline.

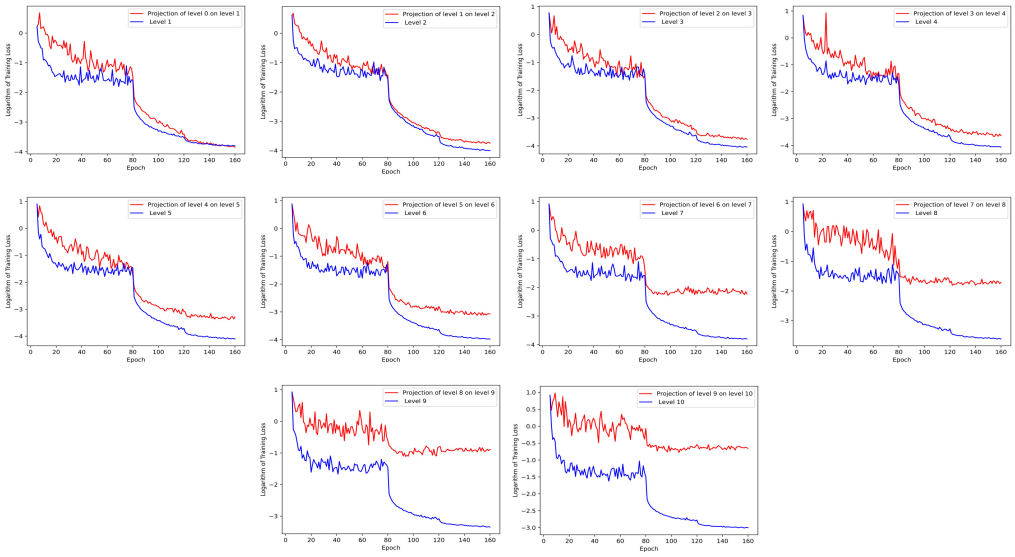

Fig. 4. Comparison of logarithm of training loss versus epoch between level (L) and level $(L-1)$ projected on level (L).

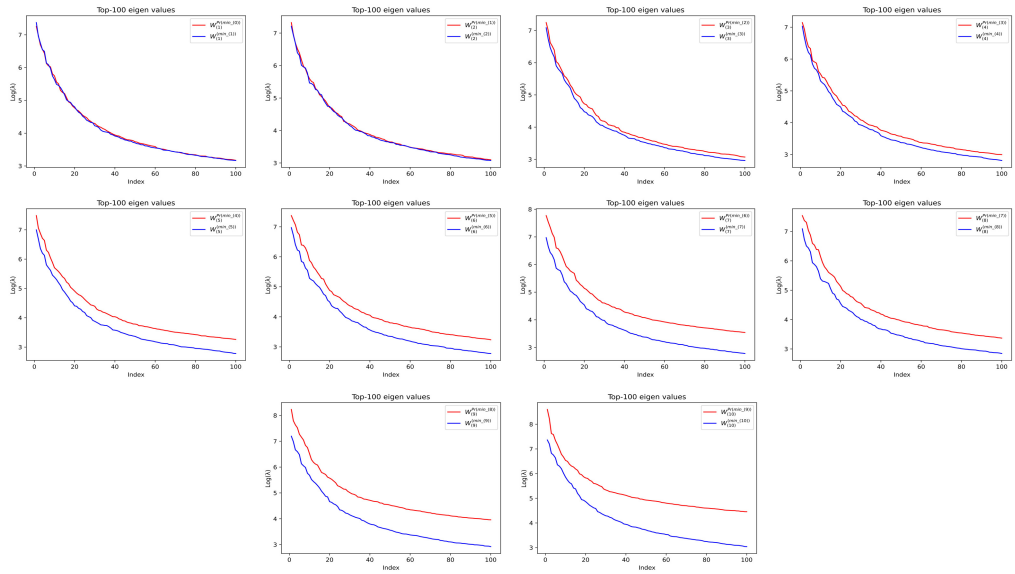

Fig. 5. Comparison of top-100 positive eigenvalues of the Hessian at $W_{(L)}^{(min\_(L))}$ and $W_{(L)}^{Pr(min\_(L-1))}$. The figure shows that the eigenvalues of the Hessian at $W_{(L)}^{(min\_(L))}$ are smaller than those at $W_{(L)}^{Pr(min\_(L-1))}$.

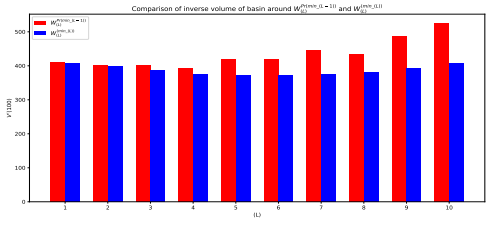

Fig. 6. Comparison of inverse volume of basin, $V^{'}(100)$ at $W_{(L)}^{Pr(min\_(L-1))}$ and $W_{(L)}^{(min\_(L))}$.

pruned minimum, thus not offering any improvement. One-shot pruning removes too many weights and therefore, has inferior performance. A comparison of top-100 positive eigenvalues of the Hessian between $W_{(10)}^{(one\_shot)}$ and $W_{(10)}^{(min\_(10))}$ (Figure 10) shows that $W_{(10)}^{(one\_shot)}$ has larger eigen-values than $W_{(10)}^{(min\_(10))}$, which implies a smaller volume for the basin around $W_{(10)}^{(one\_shot)}$. This brings out and confirms the importance of the iterative process as well as using a proper weight threshold for pruning in the iterative stages .

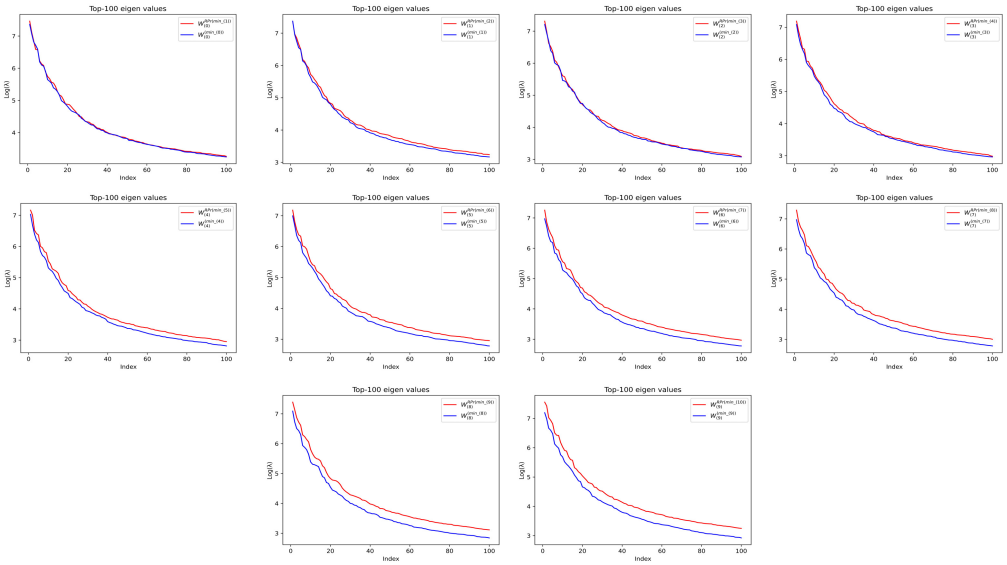

Fig. 7. Comparison of top-100 positive eigenvalues of the Hessian at $W_{(L-1)}^{(min\_(L-1))}$ and $W_{(L-1)}^{RPr(min\_(L))}$. The figure shows that the eigenvalues of the Hessian at $W_{(L-1)}^{(min\_(L-1))}$ are smaller than that at $W_{(L-1)}^{RPr(min\_(L))}$.

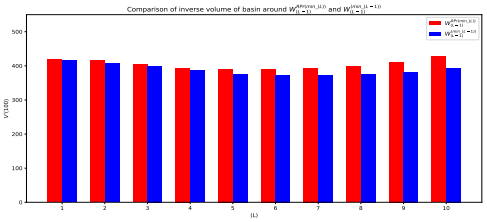

Fig. 8. Comparison of $V^{'}(100)$ at $W_{(L-1)}^{RPr(min\_(L))}$ and $W_{(L-1)}^{(min\_(L-1))}$.

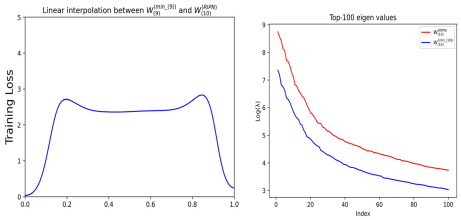

Fig. 9. **Left:** Training loss along a straight line between $W_{(9)}^{(min\_(9))}$ and $W_{(10)}^{(RIPN)}$. **Right:** Comparison of top-100 positive eigenvalues of the Hessian at $W_{(10)}^{(RIPN)}$ and $W_{(10)}^{(min\_(10))}$.

D. Result 4: There exists a barrier between IMP solutions at successive levels in the loss landscape.

One interpretation of Result 1 is that IMP solutions at different levels are distinct minima. In order to test this hypothesis more definitively, we examine whether these are separated by distinct barriers in the loss landscape. If this was not the case, SGD should converge to a sparser solution in the first place and would not stop at a less sparse solution. Figure 11 shows the training loss along a straight line connecting

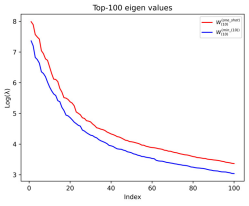

Fig. 10. Comparison of top-100 positive eigenvalues of the Hessian at $W_{(10)}^{(one\_shot)}$ and $W_{(10)}^{(min\_(10))}$.

$W_{(L-1)}^{(min\_(L-1))}$ and $W_{(L)}^{(min\_(L))}$ for different values of L and the barriers between successive minima (IMP solutions) are clearly visible. Thus, at least in the loss landscape, the IMP solutions at successive levels are not linearly connected, unlike the corresponding claim by [19] in the test error landscape. The implication is that these loss barriers are large enough to prevent SGD from crossing over and small enough to have almost similar generalization performance all over the region between these solutions.

E. Result 5: IMP solutions obtained using rewinding lie within the same loss sub-level set.

Any random initialization of a dense neural network makes SGD converge to a good minimum. This can be ascribed to the well-understood fact that the density of good minima in such scenarios is very high [29]. However, in sparse subspaces, the density of such minima is much smaller. If we consider starting from a random point in such subspaces, the solution would not be as good as we get with rewinding, as is observed in our experiments. We hypothesize that all the IMP solutions obtained using rewinding lie within the same loss sub-level

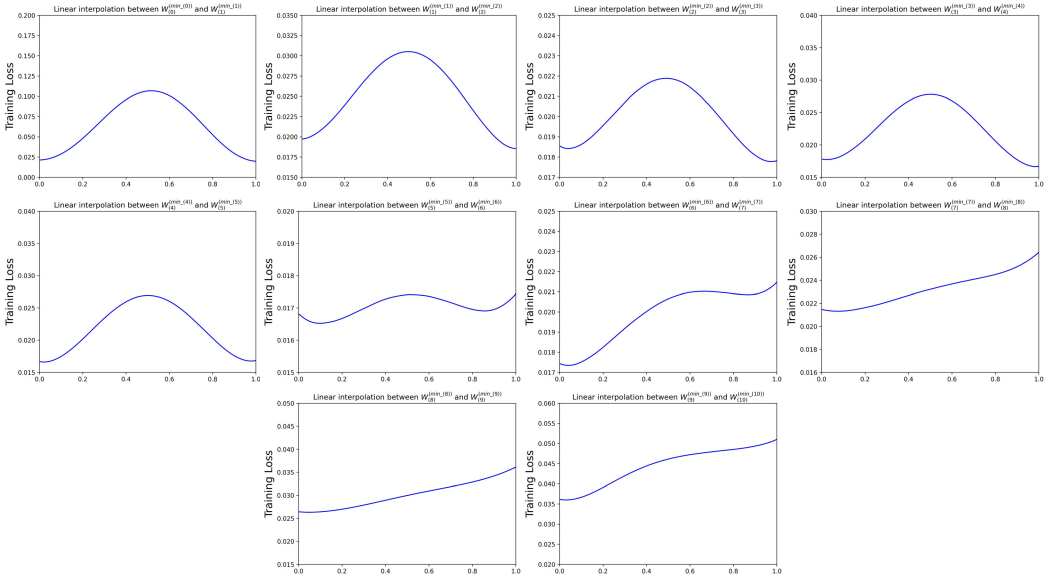

Fig. 11. Training Loss along a straight line connecting $W_{(L-1)}^{(min\_(L-1))}$ and $W_{(L)}^{(min\_(L))}$. The x-axis represents the interpolation co-efficient α. Each plot depicts the training loss at 501 points between $W_{(L-1)}^{(min\_(L-1))}$ and $W_{(L)}^{(min\_(L))}$.

set as that of the original dense network, whereas random initialization of the pruned network takes SGD out of that sub-level set. Figure 12 presents the evidence for our claim. The figure clearly shows that all the IMP solutions ($W_{(0)}^{(min\_(0))}$ to $W_{(10)}^{(min\_(10))}$) lie within the same connected loss sub-level set (dark blue region). However, $W_{(10)}^{(RIPN)}$ lies outside the sub-level set. Similarly, random pruning of a large number of weights also takes SGD outside the sub-level set. It is interesting to see that $W_{(10)}^{(one\_shot)}$ also lies in the same sub-level set as the IMP solutions; however, $W_{(10)}^{(RPN\_2)}$ lies outside the sub-level set. In other words, a randomly pruned network obtained by pruning a large number of parameters does not yield an equally good solution as that of the network obtained with magnitude based pruning, as it is not seen to lie in the same sub-level set. This can be further confirmed by calculating the cosine similarities (angular distances) between different points of interest. Figure 13 shows that the cosine similarity between IMP solutions is greater than the cosine similarity of $W_{(10)}^{(RIPN)}$ with $W_{(9)}^{(min\_(9))}$ and $W_{(10)}^{(min\_(10))}$. It also shows that the cosine similarity between $W_{(0)}^{(min\_(0))}$ and $W_{(10)}^{(one\_shot)}$ is greater than the cosine similarity of $W_{(0)}^{(min\_(0))}$ with $W_{(10)}^{(RPN\_2)}$.

F. Result 6: Why fine-tuning does not perform at par with rewinding?

Fine-tuning perturbs the pruned baseline by a small amount; hence, SGD more likely stays near the baseline solution and does not explore the minima outside the baseline. However, rewinding takes the SGD out of the baseline minimum and is more likely to converge to a better minimum in the pruned space (which was undiscoverable in the original space). Figure

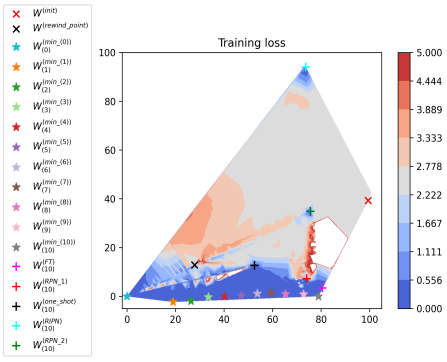

Fig. 12. The figure shows the training loss at and around the neighborhood of the points of interest. It shows that all the IMP-WR solutions lie int the same connected sub-level set (dark-blue region) . The plot has been obtained by calculating training loss at 4200 points in the high-dimensional space and then projecting the points onto the two-dimensional plane spanned by 3 points, $W_{(0)}^{(min\_(0))}$, $W_{(10)}^{(min\_(10))}$ and $W_{(10)}^{(RIPN)}$. This method of visualization is based on [28]

14 presents the comparison of top-100 positive eigenvalues of the Hessian at $W_{(10)}^{(FT)}$ and $W_{(10)}^{(min\_(10))}$. It can be observed from the figure that $W_{(10)}^{(FT)}$ has larger Hessian eigenvalues than $W_{(10)}^{(min\_(10))}$, which implies a smaller volume for the basin around $W_{(10)}^{(FT)}$. This shows the importance of rewinding the learning rate schedule while re-training the pruned network.

VI. Conclusion and Scope for Further Work

This work provides important insights into the operation of lottery ticket initializations and IMP, and in particular, shows that solutions with good generalization properties exist

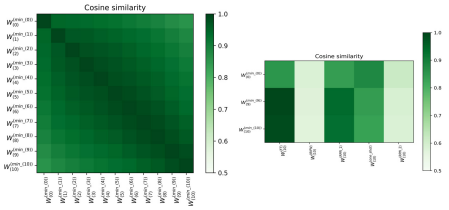

Fig. 13. Cosine similarity between different points of interest. **Left:** Cosine similarity between IMP solutions at different levels. **Right:** Cosine similarity of $W_{(10)}^{(FT)}$, $W_{(10)}^{(RIPN)}$, $W_{(10)}^{(RPN\_1)}$, $W_{(10)}^{(one\_shot)}$ and $W_{(10)}^{(RPN\_2)}$ with $W_{(0)}^{(min\_(0))}$, $W_{(9)}^{(min\_(9))}$ and $W_{(10)}^{(min\_(10))}$.

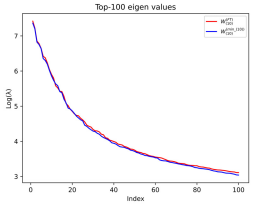

Fig. 14. Comparison of top-100 positive eigen values of the Hessian at $W_{(10)}^{(FT)}$ and $W_{(10)}^{(min\_(10))}$.

which have special geometrical profiles with low volumes in unpruned space and get discovered post-pruning. These solutions have (relatively) narrower profiles along dimensions which eventually get pruned out. There may exist solutions with similar narrow profiles along directions which do not align with a sparser weight space, which, while not important from the perspective of pruning may have other useful applications/properties and can explored in a future work. Another direction for further work is to come up with an approach that uncovers the sparse-dimensioned special solutions directly without going through the computationally and time-intensive IMP process.

ACKNOWLEDGEMENT

This work was supported by funding from IIT Delhi, Cadence Design Systems, and the Indian National Science Academy (INSA).

REFERENCES

[1] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," *arXiv preprint arXiv:1803.03635*, 2018.

[2] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" *Proceedings of machine learning and systems*, vol. 2, pp. 129–146, 2020.

[3] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in neural information processing systems*, vol. 28, 2015.

[4] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.

[5] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin, "Stabilizing the lottery ticket hypothesis," *arXiv preprint arXiv:1903.01611*, 2019.

[6] J. Frankle and D. Bau, "Dissecting pruned neural networks," *arXiv preprint arXiv:1907.00262*, 2019.

[7] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6541–6549.

[8] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin, "Pruning neural networks at initialization: Why are we missing the mark?" *arXiv preprint arXiv:2009.08576*, 2020.

[9] H. Zhou, J. Lan, R. Liu, and J. Yosinski, "Deconstructing lottery tickets: Zeros, signs, and the supermask," *Advances in neural information processing systems*, vol. 32, 2019.

[10] J. Frankle, G. K. Dziugaite, D. Roy, and M. Carbin, "Linear mode connectivity and the lottery ticket hypothesis," in *International Conference on Machine Learning*. PMLR, 2020, pp. 3259–3269.

[11] B. W. Larsen, S. Fort, N. Becker, and S. Ganguli, "How many degrees of freedom do we need to train deep networks: a loss landscape perspective," *arXiv preprint arXiv:2107.05802*, 2021.

[12] S. Zhang, M. Wang, S. Liu, P.-Y. Chen, and J. Xiong, "Why lottery ticket wins? a theoretical perspective of sample complexity on sparse neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 2707–2720, 2021.

[13] J. S. Rosenfeld, J. Frankle, M. Carbin, and N. Shavit, "On the predictability of pruning across scales," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9075–9083.

[14] R. Movva, J. Frankle, and M. Carbin, "Studying the consistency and composability of lottery ticket pruning masks," *arXiv preprint arXiv:2104.14753*, 2021.

[15] M. Paul, B. Larsen, S. Ganguli, J. Frankle, and G. K. Dziugaite, "Lottery tickets on a data diet: Finding initializations with sparse trainable networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 18 916–18 928, 2022.

[16] T. Jin, M. Carbin, D. M. Roy, J. Frankle, and G. K. Dziugaite, "Pruning's effect on generalization through the lens of training and regularization," *arXiv preprint arXiv:2210.13738*, 2022.

[17] Z. Zhang, J. Jin, Z. Zhang, Y. Zhou, X. Zhao, J. Ren, J. Liu, L. Wu, R. Jin, and D. Dou, "Validating the lottery ticket hypothesis with inertial manifold theory," *Advances in neural information processing systems*, vol. 34, pp. 30 196–30 210, 2021.

[18] B. Liu, Z. Zhang, P. He, Z. Wang, Y. Xiao, R. Ye, Y. Zhou, W.-S. Ku, and B. Hui, "A survey of lottery ticket hypothesis," *arXiv preprint arXiv:2403.04861*, 2024.

[19] M. Paul, F. Chen, B. W. Larsen, J. Frankle, S. Ganguli, and G. K. Dziugaite, "Unmasking the lottery ticket hypothesis: What's encoded in a winning ticket's mask?" *arXiv preprint arXiv:2210.03044*, 2022.

[20] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," *arXiv preprint arXiv:1810.05270*, 2018.

[21] A. Renda, J. Frankle, and M. Carbin, "Comparing rewinding and fine-tuning in neural network pruning," *arXiv preprint arXiv:2003.02389*, 2020.

[22] C. Liu, L. Zhu, and M. Belkin, "Toward a theory of optimization for over-parameterized systems of non-linear equations: the lessons of deep learning," *arXiv preprint arXiv:2003.00307*, 2020.

[23] Y. Cooper, "The loss landscape of overparameterized neural networks," *arXiv preprint arXiv:1804.10200*, 2018.

[24] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural computation*, vol. 9, no. 1, pp. 1–42, 1997.

[25] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina, "Entropy-sgd: Biasing gradient descent into wide valleys," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2019, no. 12, p. 124018, 2019.

[26] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2016.

[27] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, "Sharp minima can generalize for deep nets," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1019–1028.

[28] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," *Advances in neural information processing systems*, vol. 31, 2018.

[29] W. R. Huang, Z. Emam, M. Goldblum, L. Fowl, J. K. Terry, F. Huang, and T. Goldstein, "Understanding generalization through visualizations," 2020.

[30] L. Wu, Z. Zhu *et al.*, "Towards understanding generalization of deep learning: Perspective of loss landscapes," *arXiv preprint arXiv:1706.10239*, 2017.